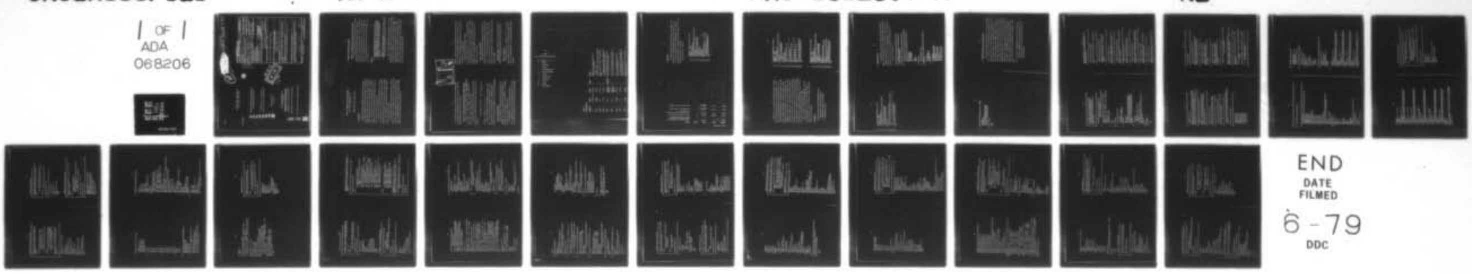
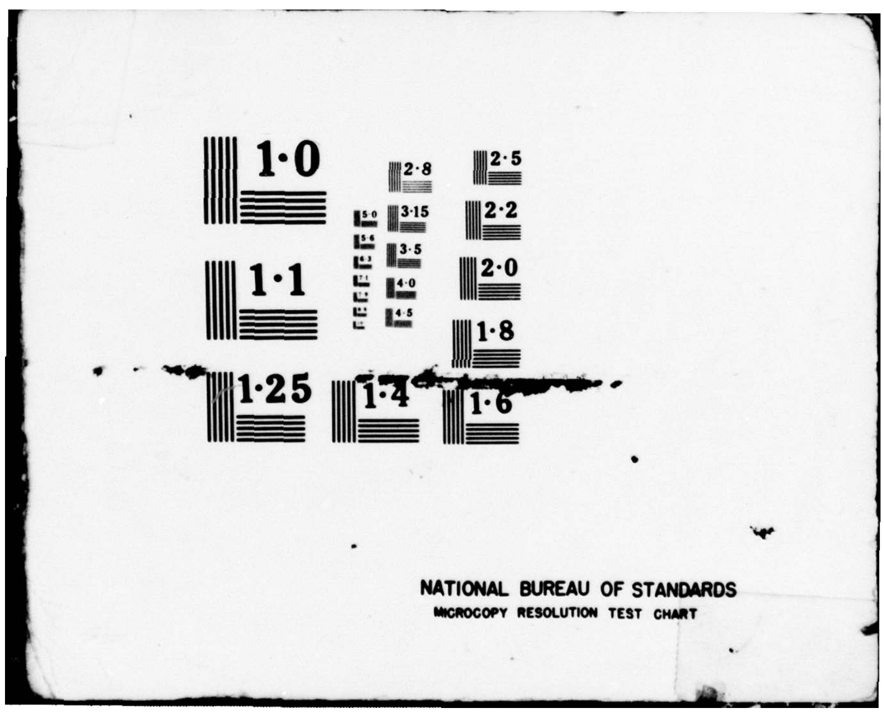
AD-A068 206 TEXAS A AND M UNIV COLLEGE STATION INST OF STATISTICS F/G 12/1
ONESAM, A COMPUTER PROGRAM FOR NONPARAMETRIC DATA ANALYSIS AND --ETC(U)
MAR 79 E PARZEN, J M WHITE DAAG29-78-G-0180
UNCLASSIFIED TR-A-7 ARO-16228.7-M NL

1 OF 1
ADA
068206



END
DATE
FILMED
6-79
DDC


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
5·0
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

[redacted] 18 ARO 19 16228.7-M

BEST AVAILABLE

DDC FILE COPY

AD A068206

79 04 30 062

TEXAS A&M UNIVERSITY
COLLEGE STATION, TEXAS 77843

INSTITUTE OF STATISTICS
Phone 713 - 845-3141

ONESAM, A COMPUTER PROGRAM FOR NONPARAMETRIC DATA ANALYSIS AND GOODNESS OF FIT

by Emanuel Parzen and J. Michael White

Institute of Statistics, Texas A&M University

Technical Report No. A-7

March 1979

Texas A & M Research Foundation
Project No. 3861

"Maximum Robust Likelihood Estimation and Non-parametric Statistical Data Modeling"

Sponsored by the U.S. Army Research Office

Professor Emanuel Parzen, Principal Investigator

Approved for public release; distribution unlimited.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Technical Report No. A-7	2. GOVT ACCESSION NO.
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): ONESAM, A Computer Program for Nonparametric Data Analysis and Goodness of Fit	5. TYPE OF REPORT & PERIOD COVERED: Technical rept.
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Emanuel Parzen and J. Michael White	8. CONTRACT OR GRANT NUMBER(s): DAAG29-78-G-0180
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Texas A&M University, Institute of Statistics, College Station, TX 77843	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Army Research Office, Research Triangle Park, NC 27709	12. REPORT DATE: March 1979
	13. NUMBER OF PAGES: 46
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited.	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report): NA	
18. SUPPLEMENTARY NOTES: The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number): Statistical data modeling, statistical computing, goodness of fit tests	
20. ABSTRACT (Continue on reverse side if necessary and identify by block number): This report is a Users' Guide to ONESAM, a mainline Fortran computer program that implements the one-sample non-parametric data analysis and Goodness of Fit procedures introduced by Parzen (1979).	

6 10 15 9 11 12 24p. 14 TR-A-7

347 380 Gu

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ONESAM, A COMPUTER PROGRAM FOR NONPARAMETRIC DATA ANALYSIS AND GOODNESS OF FIT

by

Emanuel Parzen and J. Michael White

1. Introduction

ONESAM (one sample) is a FORTRAN mainline program that implements the one sample non-parametric data analysis and Goodness of Fit techniques proposed by Parzen (1979). There are 5 stages to a quantile function and density-quantile function analysis of a single data sample. The program: Stage 1. Accepts grouped or individual (ungrouped) data, and forms non-parametric raw estimates of the quantile function; Stage 2. Forms a raw density-quantile function and sample spacings.

If the user specifies a null hypothesis probability law, the program checks the goodness of fit of the data to that probability law, and forms non-parametric smooth estimates of the density-quantile function in Stages 3 and 4. Stage 5 forms $\hat{Q}(u)$, a smooth estimator of the quantile function when a parametric model is assumed.

Section 2 describes the stages of analysis in more detail. Section 3 details the procedure to use ONESAM including the options available to users. In Section 4, several examples of the JCL used to run the jobs are given. Section 5 discusses Quantile-Box Plots, and Section 6 introduces a program QANHAT for parametric estimation of location and scale.

2. Comments on the Stages of Analysis

Stage 1: User inputs data, outputs sample quantile function.

A. If the input is ungrouped data, the user must supply: the tape number (specified by DATA), upon which the data resides (if other than unit 5 which is the tape the data is usually on when read in from cards); the format of the data; and the data itself. The output of stage 1 consists of: various descriptive statistics; the order statistics by quartiles; and the empirical quantile function, $\tilde{Q}(u)$, defined by $\tilde{Q}(u) = X_{j;n}$ for $u = \frac{j-.5}{n}$, $j = 1, \ldots, n$.

Options:

1. Data can be standardized to take values in the unit (0, 1) interval by specifying IOPT2 = 1. The output (which includes descriptive statistics, order statistics by quartiles, and $\tilde{Q}$) is based on the standardized data. When this option is used, future stages of analysis will use standardized data for analysis.

2. Compute $\tilde{Q}(u)$ using linear interpolation, for $u = jh$, $j = 1, \ldots, NQ$ where $NQ = [1/h]$ by specifying FINC = h. When this option is used, future stages will use $\tilde{Q}$ computed at points h, 2h, ..., 1 - h.

B. If the user inputs grouped data by specifying IOPT1 = 1, the user must also supply: the lower limit of the first interval (XMIN); the interval width (XINT); and, if desired, the total number of observations (NOBS). The user must input the numbers of observations in each interval in Format (20F4.0) and supply a value, h (FINC), to be used in computing $\tilde{Q}$. As with the raw data, the user must supply the number of the unit (DATA) that the data is to be read from (if other than unit 5).

Also available in Stage 1 is the Quantile Box Plot procedure which is not part of the ONESAM package but is easy to access by a short user-supplied FORTRAN program described in Section 5.

<u>Stage 2</u>: <u>Raw density-quantile function $\tilde{f}Q(u)$ and sample spacings.</u>

In Stage 2, the function $\tilde{f}Q(u)$, defined by $\tilde{f}Q(jh) = \dfrac{2h}{\tilde{Q}((j+1)h)-\tilde{Q}((j-1)h)}$ for $u = jh$, $j = 1, \ldots, NQ$, is computed. Also computed in Stage 2 is $\tilde{q}(u) = 1/\tilde{f}\tilde{Q}(u)$. Plots of these functions are given.

Note that if $\tilde{Q}$ is not computed via linear interpolation from grouped data, but rather is computed from ungouped data, the definition of $\tilde{q}(u)$ is such that

$$\tilde{q}(u) = n\{\tilde{Q}((j+.5)/n) - \tilde{Q}((j-.5)/n)\} = n\{X_{(j+1);n} - X_{j;n}\}$$

for $\frac{j-.5}{n} \le u < \frac{j+.5}{n}$, $j = 1, 2, \ldots, n-1$.

Stages 1 and 2 constitute the non-parametric raw analysis of the data. If the user supplies a null-hypothesis probability law, one can proceed to Stages 3 and 4 which constitute a non-parametric smooth analysis of the data and also a goodness of fit test for the null hypothesis.

<u>Stage 3</u>: <u>Raw $\tilde{D}(u)$ and $\tilde{\phi}(v)$.</u>

The user must specify a density-quantile function f_0Q_0 for the null hypothesis H_0: $Q = \mu + \sigma Q_0$. This is done using indicator variables. If one wishes to test against density "j", then set DQH(j) = 1. The possible densities one can use are supplied in Table I.

The program computes: (1) the raw transformation density, $\tilde{d}(u) = f_0Q_0(u)\tilde{q}(u)/\tilde{\sigma}_0$, also called the weighted spacings; and (2) the raw transformation distribution function, $\tilde{D}(u)$ called the cumulative weighted

spacings. (Tests for H_0 could be obtained by using the facts that under H_0 the $\tilde{d}$'s should be distributed as Uniform (0, 1), and $\sqrt{n}\,\{\tilde{D}(u) - u\}$ is asymptotically a Brownian Bridge stochastic process). $\tilde{\sigma}_0$ is an estimate of σ_0 defined by $\sigma_0 = \int_0^1 f_0Q_0(u)q(u)du$.

The program then computes the Fourier transform $\tilde{\phi}(v) = \int_0^1 \tilde{d}(u)\exp(2\pi ivu)du$ for $v = 0, 1, \ldots, m$ by specifying ORDER = m (the program chooses order 5 if no other order is specified). By comparing the square modulus of the phi's, $|\tilde{\phi}(v)|^2$, which are plotted, to a suitable threshold value (for example, 2*NOBS), one could use these values as evidence to accept or reject H_0.

<u>Stage 4</u>: <u>Smooth $\hat{D}(u)$ and $\hat{f}Q(u)$.</u>

Autoregressive smoothed density estimators $\hat{f}Q$ and smoothed transformation distribution functions $\hat{D}(u)$ are computed for successive orders 1, 2, ..., m. Plots of $\tilde{D}$ vs $\hat{D}$ are given for all orders to help select the order that gives the best fit to the data. Also given is the CAT criterion. If CAT selects order 0, one uses this as evidence to not reject the null hypothesis.

<u>Alternative analyses</u>: Stages 3 and 4 can be repeated for several possible hypothesized densities simply by setting any other DQH(j) = 1.

<u>Stage 5</u>: <u>Smooth $\hat{Q}(u)$.</u>

The program computes $\hat{Q}(u) = \hat{\mu} + \hat{\sigma}Q_0(u)$, $u = jh$, $j = 1, \ldots, NQ$ for several estimates $\hat{\mu}$, $\hat{\sigma}$ of μ and σ. This procedure is available but is not part of the ONESAM package; it is a separate program QANHAT described in Section 6.

TABLE I

List of Density-Quantile Functions

j	f_0Q_0
1	Normal
2	Exponential
3	Logistic
4	Double Exponential
5	Uniform Reciprocal
6	Cauchy
7	Extreme Value
8	Log Normal
9	Pareto
10	Weibull
11	Half Logistic

TABLE II

Parameter List

Parameter	Value	Status	Default	Remarks
N =	integer ≤ 510	required	0	Sample size for raw data; the number of intervals for grouped data
DATA =	integer	optional	5	1 = data on unit 1 (if Data = 2, card 14 must change accordingly)
TRANS =	real	optional	1.0	Value of exponent for power transformation if desired
IOPT1 =	0, 1, 2	optional	0	1 for grouped data with equal interval widths 2 for grouped data in unequal intervals
XMIN =	real	optional required if IOPT1 = 1	first order statistic	Natural minimum of data. For grouped data, it is lower limit of first interval
FINC =	real	optional required if IOPT1 = 1	0.0	Value of h for computing equally spaced $\tilde{Q}$. If used, FINC usually equals .01, .025, or .05
NOBS =	integer	optional	N	Number of observations for grouped data. It is used in computing CAT
IOPT2 =	0, 1	optional	0	1 if user desires data to be standardized to (0, 1) before analysis. Sometimes useful for comparing several batches
ORDER =	integer	optional	5	Maximum order to be used for autoregressive estimator usually ≤ 10

TABLE II (continued)

Parameter	Value	Status	Default	Remarks
DQN(J) = J = 1, ... 12	0, 1	optional	0	1 if user wishes to test against "J"th hypothesized density. See table in Section 3 for allowable densities
WIDTH =	integer	optional	80	Number of columns in printer plots
PLOT =	0, 1	optional	0	1 if CAT comp plots desired (unavailable at present time)
BATAP =	real > 0	required if D(9) = 1	--	Shape parameter for Pareto density
BETAW =	real > 0	required if D(10) = 1	--	Shape parameter for Weibull density

3. Procedures

This section details the procedure one should use to complete a data analysis using the ONESAM package and gives the JCL necessary to run a job. The most critical feature of the ONESAM package is to understand completely the parameter list PARMS for this contains almost all the information and options used in the analysis.

A typical JCL deck of cards is:

```
1.   //XXXXX JOB (XXXX,XXXX,$30,008,XX),'XXXXXX'
2.   /*PASSWORD     XXXXX
3.   /*JOBPARM R=192,K=0
4.   // EXEC FORTXCG,FXREGN=192K
5.   //FORT.SYSIN DD DSN=UYL.MJ.ENP.QUANT(ONSPL1),DISP=SHR
6.   //GO.SYSLIB DD
7.   //  DD DSN=UYL.MJ.ENP.COSUB1,DISP=SHR
8.   //  DD DSN=UYL..JO.ELH.CTSBD,DISP=SHR
9.   //GO.SYSIN DD *
10.  &A NBATCH=III,&END
11.  DATA SET NAME
12.  &PARMS N=III,              ,&END
13.  FORMAT OF DATA EG (16F5.1)
14.  //GO.FT01F001 DD DSN=UYL.XX.XXX.XXXXX(XXXXXX),
15.  // DISP=SHR,LABEL=(,,,IN)
16.  //GO.FT09F001 DD UNIT=SYSDA,SPACE=(TRK,50),
17.  // DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
18.  /*END
```

Cards 10, 12, 13, 14, and 15 require further explanation. Cards 10 and 12 utilize the name list feature of the FORTRAN language. Card 10 tells the program how many batches are being analyzed in this run. If only one batch is being analyzed, the card should be punched: b&A &END (Note: b indicates a blank space.) If more than one batch is being analyzed, the user would repeat cards 11 through 15 NBATCH times. The user <u>must</u> put a blank in column 1 of card 10 and an ampersand in column 2. Card 12 is the parameter card. The user must put a blank in column 1 of card 12 and an & in column 2. Columns 3-7 contain the word PARMS followed by a blank in column 8. The rest of the card contains the desired parameters selected from the parameter list in Table II. Some of the parameters are required, some are optional as explained belowed. They can be listed in any order. The parameter list must end with ,&END.

Card 13 is required if the user inputs ungrouped data. Omit card 13 if using grouped data. Cards 14 and 15 are used if the data is located on an external field. If the data is on cards, replace cards 14 and 15 with the data cards.

4. Sample JCL

Sample JCL are given for the following types of jobs:

A. Ungrouped data on cards, with three batches;

B. Ungrouped data on external file (Wylbur);

C. Grouped data on cards;

A

```
1.   //WHITE JOB (E483,002D,S30,005,MU),'BUFSNO'
2.   /*PASSWORD     XXXXX
3.   /*JOBPARM R=192,K=0
4.   // EXEC FORTXCG,FXREGN=192K
5.   //FORT.SYSIN DD DSN=WYL.MU.ENP.QUANT(ONSPL1),DISP=SHR
6.   //GO.SYSLIB DD
7.   // DD DSN=WYL.MU.ENP.COSUB1,DISP=SHR
8.   // DD DSN=WYL.JO.ELH.CTSDD,DISP=SHR
9.   //GO.SYSIN DD *
10.   &A NBATCH=3,&END
11.  TIPPETT'S WARP BREAKS  AL
12.   &PARMS N=9,DOH(1)=1,DOH(2)=1,&END
13.  (10F5.0)
14.    26.  30.  54.  25.  70.  52.  51.  26.  67.
15.  TIPPETT'S WARP BREAKS  AM
16.   &PARMS N=9,DOH(1)=1,DOH(2)=1,&END
17.  (10F5.0)
18.    18.  21.  29.  17.  12.  18.  35.  30.  36.
19.  TIPPETT'S WARP BREAKS  AH
20.   &PARMS N=9,DOH(1)=1,DOH(2)=1,&END
21.  (10F5.0)
22.    36.  21.  24.  18.  10.  43.  28.  15.  26.
23.  //GO.FT09F001 DD UNIT=SYSDA,SPACE=(TRK,50),
24.  // DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
25.  /*END
```

B

```
1.   //WHITE JOB (E483,002D,S30,005,MU),'BUFSNO'
2.   /*PASSWORD     XXXXX
3.   /*JOBPARM R=192,K=0
4.   // EXEC FORTXCG,FXREGN=192K
5.   //FORT.SYSIN DD DSN=WYL.MU.ENP.QUANT(ONSPL1),DISP=SHR
6.   //GO.SYSLIB DD
7.   // DD DSN=WYL.MU.ENP.COSUB1,DISP=SHR
8.   // DD DSN=WYL.JO.ENH.CTSDD,DISP=SHR
9.   //GO.SYSIN DD *
10.   &A &END
11.  BUFFALO SNOWFALL DATA (1910-1972)
12.   &PARMS N=63,DOH(1)=1,DATA=1,FINC=.05,&END
13.  (5F10.0)
14.  //GO.FT01F001 DD DSN=WYL.MU.ENP.QUANT(BFSND),
15.  // DISP=SHR,LABEL=(,,,IN)
16.  //GO.FT09F001 DD UNIT=SYSDA,SPACE=(TRK,50),
17.  // DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
18.  /*END
```

```
C

//WRITE JOB (E483,002D,S30,005,MB),'BPDATA'
/*PASSWORD     XXXXX
/*LCSPARM R=192,K=0
// EXEC FORTXCG,FXREGN=192K
//FORT.SYSIN DD DSN=UYL.MJ.ENP.QUANT(ONSPL1),DISP=SHR
//GO.SYSLIB DD
// DD DSN=UYL.MU.ENP.COSUBS,DISP=SHR
// DD DSN=UYL.JD.ELN.CTSBD,DISP=SHR
//GO.SYSIN DD *
 $A $END
BLOOD PRESSURE DATA--NON-USERS--AGE 17-24
 $PARMS N=15,D(1)=1,DATA=5,IOPT1=1,XMIN=85.,XINT=5.,FINC=.05,
 $END
   0   1   3  10  11  15  20  13  10   8   4   3   2   0   0   0
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(TRK,50),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
/*END
```


5. Quantile Box-Plot

A simple FORTRAN driver program to access the subroutines QDIAG and QBOX to perform the Quantile Box plot procedures and compute the associated diagnostics is supplied below. If the user desires a Calcomp plot of the Quantile Box Plot, he can use the subroutine QBXPLT. Be cautioned that the data vectors and vector of "u-values" must be dimensioned at least 2 * N + 1 where N is the sample size.

```
C**********************************************************
C
C   THIS PROGRAM IS THE DRIVER FOR THE QUANTILE BOX PLOT
C   ANALYSIS AND DIAGNOSTICS.
C
C**********************************************************
C
      DIMENSION DATA(200),X1(200),X2(200),Y(200),U(200),LABEL(20)
C
      DATA LABEL/4HBUFF,4HALO ,4HSNOW,4HFALL,4H 191,4H0-72,
     +14*4H    /
      DATA LAB1/4HORIG/
      DATA LAB2/4HSORT/
      DATA LAB3/4HLOG /
C
      N=63
      NN1=2*N+1
      IOPT=0
C
C  OBTAIN DATA
C
      READ(5,100)(DATA(I),I=1,N)
C
C  COMPUTE PRELIMINARY TRANSFORMATIONS OF DATA FOR ANALYSIS
C  IF DESIRED
C
      DO 10 J=1,N
      X1(J)=SQRT(DATA(J))
      X2(J)=ALOG(DATA(J))
   10 CONTINUE
C
      LABEL(8)=LAB1
      CALL QDIAG(DATA,N,IOPT,LABEL)
      CALL QBOX(DATA,Y,NN1,LABEL)
      LABEL(8)=LAB2
      CALL QDIAG(X1,N,IOPT,LABEL)
      CALL QBOX(X1,Y,NN1,LABEL)
      LABEL(8)=LAB3
      CALL QDIAG(X2,N,IOPT,LABEL)
      CALL QBOX(X2,Y,NN1,LABEL)
```

```
C
C     OBTAIN CALCOMP OF QUANTILE BOX PLOT, IF DESIRED
C
      LABEL(8)=4H
      LAB=10HORIGINAL
      CALL QBXPLT(DATA,NN1,LABEL,3,U,2,LAB1)
      CALL QBXPLT(X1,NN1,LABEL,3,U,2,LAB2)
      CALL QBXPLT(X3,NN1,LABEL,3,U,2,LAB3)
C
  100 FORMAT(8F10.0)
C
      STOP
      END
```


6. Parametric Estimation of Location and Scale

One can obtain parametric estimates $\hat{\mu}$ and $\hat{\sigma}$ of μ and σ in the model $Q(u) = \mu + \sigma Q_0(u)$. For a complete discussion of the derivation of $\hat{\mu}$ and $\hat{\sigma}$ see Parzen (1978). At the present time parametric estimates are available only when we assume $Q_0 = \Phi^{-1}$. Estimates $\hat{\mu}(p)$ and $\hat{\sigma}(p)$ can be computed using the subroutine STEP1, MUHAT, and SIGHAT. The variance of each estimator is also computed. The estimate $\hat{Q}_p(u) = \hat{\mu}(p) + \hat{\sigma}(p)\Phi^{-1}(u)$ can be computed using the estimates $\hat{\mu}(p)$ and $\hat{\sigma}(p)$ together with subroutine MDNRIS which will compute $\Phi^{-1}(u)$. A plot of $\hat{Q}(u)$ vs $\tilde{Q}(u)$ would be useful to check the fit of the parametric model and can be obtained using subroutine JPPLOT. A sample FORTRAN program to produce the estimates and desired plots is given below. This sample program uses ungrouped data as input and would have to be modified somewhat to accomodate grouped data.

Note: This stage of the analysis is in the process of revision and expansion to accommodate the full range of Q_0 functions. Provisions to make this stage of analysis more accessible to users are being developed.

```
C*************************************************************
C
C     THIS PROGRAM IS THE DRIVER TO OBTAIN ESTIMATES
C     MUHAT(P) AND SIGHAT(P) FOR VARIOUS VALUES OF P.
C     IT ALSO COMPUTES QHAT(U) AND QTILDE(U) FOR
C     UNGROUPED DATA, AND PLOTS QHAT AND QTILDE ON THE
C     SAME SET OF AXES
C
C*************************************************************
C
      DIMENSION DATA(200),U(200),QHAT(200),PHI(200),
     +          LAB1(20),LAB2(20),LAB3(20),
     +          P(4),XMU(4),VMU(4),UMU(4),XSIG(4),VSIG(4),USIG(4)
C
      DATA LAB1/4HBUFF,4HALO ,4HSNOW,4HFALL,4H 191,4H0-72,
     +14*4H    /
      DATA LAB2/4HPLOT,4H OF ,4HQHAT,4H(U) ,4HVS Q,4HTILD,4HE(*),
     +13*4H    /
      DATA LAB3/4HP = ,19*4H    /
      DATA P/0.,.05,.1,.25/
C
      NP=4
      N=63
C
C     OBTAIN DATA
C
      READ(5,100)(DATA(I),I=1,N)
C
      DO 10 I=1,N
      U(I)=(FLOAT(I)-.5)/FLOAT(N)
      CALL MDNRIS(U(I),PHI(I),IER)
   10 CONTINUE
C
      WRITE(6,101)
      DO 20 IP=1,NP
      CALL STEP1(P(IP),UMU(IP),USIG(IP),VMU(IP),VSIG(IP))
      CALL MUHAT(P(IP),DATA,N,UMU(IP),VMU(IP),XMU(IP))
      CALL SIGHAT(P(IP),DATA,N,USIG(IP),VSIG(IP),XSIG(IP))
      WRITE(6,102)P(IP),XMU(IP),VMU(IP),XSIG(IP),VSIG(IP)
   20 CONTINUE
C
      DO 40 IP=1,NP
        DO 30 I=1,N
        QHAT(I)=XMU(IP)+XSIG(IP)*PHI(I)
   30   CONTINUE
      CALL FCODEA(9,P(IP),8H(F6.4)  ,LAB3(2))
      CALL JPPLOT(QHAT,U,N,80,4HQ(U),4H   U,LAB2,LAB3,6,1,1,1,
     +DATA,1H*,1,LAB1)
   40 CONTINUE
C
  100 FORMAT(
  101 FORMAT(//5X,
  102 FORMAT(
C
      STOP
      END
```



```
INDEX OF SUBROUTINES USED IN THE ONESAM PACKAGE.  UTILITY
SUBROUTINES ARE LISTED IN A SEPARATE GROUPING FROM
COMPUTING SUBROUTINES.

****************************************************************

  FUNCTION AREST(X,L,OPTKHM,OPTCOE)

FUNCTION TO COMPUTE AUTOREGRESSIVE ESTIMATOR EVALUATED
AT X.

****************************************************************

  SUBROUTINE AUTORG(A,LS,M,ALPHA,PHI,FKM)

  COMPUTES THE COEFFICIENTS ALPHA(.) AND FKM OF THE
  AUTOREGRESSIVE ESTIMATOR ACCORDING TO A RECURSIVE
  ALGORITHM

****************************************************************

  SUBROUTINE DATSTD(X,XO,N,ON)

SUBROUTINE TO STANDARDIZE VECTOR X BY (X(I)-XO)/(X(N)-XO)
AND RETURN THE STANDARDIZED DATA IN THE VECTOR ON

****************************************************************

  SUBROUTINE DESTAT(X,N,NAME,IUNIT,IHEAD)

SUBROUTINE TO PRINT ORDERED ARRAY BY QUANTILES AND COMPUTE
DESCRIPTIVE STATISTICS.

****************************************************************
  SUBROUTINE FORIER(F,U,N,A,MA)

SUBROUTINE TO COMPUTE THE FOURIER TRANSFORM
PHI(V) OF A DENSITY DEFINED ON (0,1) FOR V=0,1,....,M

****************************************************************

  SUBROUTINE KSD(D,U,N,DM,UM,DP,UP)

SUBROUTINE TO COMPUTE KOMOLGOROV-SMIRNOFF STATISTIC FOR
THE DEVIATIONS D(U)-U.  UPPER AND LOWER BOUNDS ARE COMPUTED

****************************************************************

  SUBROUTINE PARZ(RVAR,M,N,CAT,MORD)

SUBROUTINE TO DETERMINE THE ORDER OF AN AUTOREGRESSIVE
PROCESS BY PARZENS CAT CRITERIA

****************************************************************

  SUBROUTINE PRINTA(A,N,IUNIT)

SUBROUTINE TO COMPUTE AND PRINT THE SQUARE MODULUS OF THE
COMPLEX-VALUED FOURIER TRANSFORMS A(1),...,A(N)
```

```
**************************************************************

    SUBROUTINE QUICK(N,T)

  QUICK SORT.THIS ALGORITH IS ALSO REFERRED TO AS A PARTITIONED
  EXCHANGE SORT.EXPECTED RUNTIME IS PROPORTIONAL TO N*LOG2(N)
  ALTHOUGH THE WORST CASE IS PROPORTIONAL TO N**2.
  REFERENCE: DONALD E. KNUTH- THE ART OF COMPUTER PROGRAMMING VOL 3.

**************************************************************
    SUBROUTINE QHLIN(X,N,H,U,Q,IOPT,NO,XMIN,XINT)

  SUBROUTINE TO COMPUTE Q(U) AT U = H,2H,3H,....,1-H WHERE
  H IS > OR = .002 BY LINEAR INTERPOLATION FROM ORDER
  STATISTICS IF DATA IS UNGROUPED (IOPT=0), OR FROM TALLIES
  IF DATA IS GROUPED (IOPT=1).

**************************************************************

    SUBROUTINE QTOFQ(Q,U,NO,XS,FQ)

  SUBROUTINE TO COMPUTE Q(U) AND FQ(U)=1./Q(U) FROM
  THE EMPIRICAL QUANTILE FN CAP Q(U) AND THE U VALUES.

**************************************************************

    SUBROUTINE USPACE(UXS,CUXS,NO,FQ,FOQO,U,SIGO)

  SUBROUTINE TO COMPUTE D(U), CUMULATIVE D'S, AND SIGMAO
  FOR THE MODEL Q(U)=MU+SIGMA*QO(U)

**************************************************************

    SUBROUTINE ONESAM(X,N,XO,QN,XS,UXS,CUXS,IUNIT,IOPT,NAME,
   +MM,JOPT,U,DHAT,F,FQ,IFIRST,FQNAME,N,ICASE,IOPT1,IOPT2,NOBS,FINC,
   +XINT,NO)

  DRIVER FOR ONE SAMPLE ANALYSIS

**************************************************************

    FUNCTION FQCAUC(X)
    FUNCTION FQDEXP(X)
    FUNCTION FQEVAL(X)
    FUNCTION FQEXP(X)
    FUNCTION FQLGNO(X)
    FUNCTION FQLOGI(X)
    FUNCTION FQNORM(X)
    FUNCTION FQPARE(X)
    FUNCTION FQWEIB(X)
    FUNCTION FQNLOG(X)
```



```
**************************************************************

  UTILITY SUBROUTINES

**************************************************************

    SUBROUTINE ACODE(NTAPE,XNAME,IFORM,X)

  SUBROUTINE TO CONVERT 8 CHARACTER ALPHANUMERIC ARRAY
  XNAME WHICH IS IN A-FORMAT TO THE REAL VARIABLE
  X WHICH HAS THE 8 CHARACTER F-FORMAT IFORM

**************************************************************

    SUBROUTINE FCODEA(NTAPE,X,IFORM,NAME)

  SUBROUTINE TO CONVERT REAL VARIABLE X
  WHICH HAS 4 CHARACTER F-FORMAT IFORM
  TO 8 CHARACTER ALPHAMERIC ARRAY NAME WHICH IS
  IN A FORMAT.

**************************************************************

    SUBROUTINE ICODEA(NTAPE,K,IFORM,NAME)

  SUBROUTINE TO CONVERT INTEGER VARIABLE K
  WHICH HAS 8 CHARACTER I-FORMAT IFORM
  TO 8 CHARACTER ALPHAMERIC ARRAY NAME WHICH IS
  IN A-FORMAT.

**************************************************************

    SUBROUTINE JPPLOT(X,Y,N,MM,NAMEX,NAMEY,NAME,ITITLE,IUNIT,IS3,IOPT,
   +                  JSTART,Z,IZ,NOPT,IHEAD)

  SUBROUTINE TO PRINT AND PRINTER PLOT THE VECTOR X,
  LISTING J AND OPTIONALLY A VECTOR Y,WHERE J IS THE SEQUENCE. MAX 50
  (EVENLY DISTRIBUTED) VALUES ARE PLOTTED.

**************************************************************

  SUBROUTINES NOT ON THE ONESAM LIBRARY WHICH ARE USED IN THE
  ONESAM PACKAGE:

  MDNRIS : IMSL SUBROUTINE TO COMPUTE INVERSE NORMAL DISTRIBUTION
           FUNCTION

  USMNMX : IMSL SUBROUTINE TO COMPUTE MIN AND MAX OF A VECTOR

  QLPLT1 : TSBD SUBROUTINE TO PRINT AND PLOT A VECTOR (SEE JOE
           NEWTON FOR DOCUMENTATION)

**************************************************************
```

```
C
C*****************************************************************************
C
C
C   DRIVER PROGRAM FOR ONE SAMPLE ANALYSIS.
C
C
C
C*****************************************************************************
      COMMON BETAP,BETAU
      DIMENSION X(512),XS(512),UXS(512),NAME(20),IFORM(20),QN(512)
      DIMENSION  U(512)
      DIMENSION DHAT(512)
      DIMENSION CUXS(512)
      DIMENSION F(512)
      DIMENSION FQ(512)
      INTEGER DATA,WIDTH
      INTEGER DOH(30),ORDER,PLOT
      EXTERNAL FONORM
      EXTERNAL FOEXP
      EXTERNAL FOLOGI
      EXTERNAL FODEXP
      EXTERNAL FOCAUC
      EXTERNAL FOLCHO
      EXTERNAL FOPARE
      EXTERNAL FOWEIB
      EXTERNAL FOWLOG
      DATA XONE/.0000000001/
      NAMELIST/PARMS/PLOT,DATA,WIDTH,ORDER,N,TRANS,XMIN,BETAP,
     +BETAU,DOH,XINT,IOPT1,IOPT2,NOBS,FINC/A/NBATCH
      NBATCH=1
      READ(5,A)
      DO 999 IBATCH=1,NBATCH
   10 CONTINUE
      READ(5,1001) NAME
 1001 FORMAT(20A4)
      DO 15 I = 1,30
   15 DOH(I) = 0
      ORDER = 5
      WIDTH = 0
      DATA = 5
      PLOT = 0
      N = 0
      TRANS = 1.0
      XMIN = XONE
      XINT=0.
      IOPT1=0
      IOPT2=0
      NOBS=0
      FINC=0.
      READ(5,PARMS)
      WIDTH=70
C     IF(WIDTH .EQ. 0) WIDTH = (10*N)/(8*((N-1)/71+1))
      ORDER = ORDER + 1
      IF(N .LE. 511) GO TO  19
      WRITE(6,901) N
      GO TO 999
   19 CONTINUE
      IF(IOPT1.GT.0)GO TO 155
```



```
C
C   READ UNGROUPED DATA FOR IOPT1=0
C
      READ(5,1001) IFORM
      READ(DATA,IFORM) (X(I),I=1,N)
      Z = SIGN(1.0,TRANS)
      IF(TRANS .EQ. 1.0) GO TO 55
C
C   TRANSFORM UNGROUPED DATA IF DESIRED
C
      IF(TRANS .EQ. 0.0) GO TO 52
      DO 51  I = 1,N
   51 X(I) = Z*X(I)**TRANS
      WRITE(6,905)
  905 FORMAT(1H1/////)
      GO TO 55
   52 DO 53  I = 1,N
   53 X(I) = ALOG(X(I))
      WRITE(6,905)
C
   55 CONTINUE
      IFIRST = 1
      X0 = XMIN
      GO TO 57
C
C   READ GROUPED DATA IN EQUAL-SIZED INTERVALS FOR IOPT1=2
C
  155 READ(DATA,903)(X(I),I=1,N)
      X0=XMIN
      IF(IOPT1.EQ.1)GO TO 56
      READ(5,1001)IFORM
      READ(DATA,IFORM)(QN(I),I=1,N)
   56 CONTINUE
      IFIRST=1
C
C
   57 CONTINUE
      IF(DOH(1) .EQ. 0) GO TO 22
      IF(PLOT .EQ. 0)
     +CALL ONESAM(X,N,X0,QN,XS,UXS,CUXS,6,2,NAME,WIDTH,1,U,DHAT,F,FQ,
     +IFIRST,FONORM,ORDER,6HNORMAL,IOPT1,IOPT2,NOBS,FINC,XINT,NQ)
      IFIRST = 0
   22 CONTINUE
      IF(DOH(2) .EQ. 0) GO TO 23
      IF(PLOT .EQ. 0)
     +CALL ONESAM(X,N,X0,QN,XS,UXS,CUXS,6,2,NAME,WIDTH,1,U,DHAT,F,FQ,
     +IFIRST,FOEXP ,ORDER,8HEXPONTAL,IOPT1,IOPT2,NOBS,FINC,XINT,NQ)
      IFIRST = 0
   23 CONTINUE
      IF(DOH(3) .EQ. 0  ) GO TO 24
      IF(PLOT .EQ. 0)
     +CALL ONESAM(X,N,X0,QN,XS,UXS,CUXS,6,2,NAME,WIDTH,1,U,DHAT,F,FQ,
     +IFIRST,FOLOGI,ORDER, 8HLOGISTIC,IOPT1,IOPT2,NOBS,FINC,XINT,NQ)
      IFIRST = 0
   24 CONTINUE
      IF(DOH(4) .EQ. 0  ) GO TO 25
      IF(PLOT .EQ. 0)
     +CALL ONESAM(X,N,X0,QN,XS,UXS,CUXS,6,2,NAME,WIDTH,1,U,DHAT,F,FQ,
     +IFIRST,FODEXP,ORDER,7HDBL EXP,IOPT1,IOPT2,NOBS,FINC,XINT,NQ)
      IFIRST = 0
```

```
   25 CONTINUE
      IF(DOH(5) .EQ. 0  ) GO TO 26
      IFIRST = 0
   26 CONTINUE
      IF(DOH(6) .EQ. 0  ) GO TO 27
      IF(PLOT .EQ. 0)
     +CALL ONESAM(X,N,XO,ON,XS,UXS,CUXS,6,2,NAME,WIDTH,1,U,DHAT,F,FQ,
     +IFIRST,FQCAUC,ORDER,6HCAUCHY,IOPT1,IOPT2,NOBS,FINC,XINT,NQ)
      IFIRST = 0
   27 CONTINUE
      IF(DOH(7) .EQ. 0  ) GO TO 28
      IFIRST = 0
   28 CONTINUE
      IF(DOH(8) .EQ. 0  ) GO TO 29
      IF(PLOT .EQ. 0)
     +CALL ONESAM(X,N,XO,ON,XS,UXS,CUXS,6,2,NAME,WIDTH,1,U,DHAT,F,FQ,
     +IFIRST,FQLGNO,ORDER,8HLOG NORM,IOPT1,IOPT2,NOBS,FINC,XINT,NQ)
      IFIRST = 0
   29 CONTINUE
      IF(DOH(9) .EQ. 0  ) GO TO 30
      IF(BETAP .GT. 0.) GO TO 101
      WRITE(6,902) BETAP
  101 CONTINUE
      IF(PLOT .EQ. 0)
     +CALL ONESAM(X,N,XO,ON,XS,UXS,CUXS,6,2,NAME,WIDTH,1,U,DHAT,F,FQ,
     +IFIRST,FQPARE,ORDER,6HPARETO,IOPT1,IOPT2,NOBS,FINC,XINT,NQ)
      IFIRST = 0
   30 CONTINUE
      IF(DOH(10).EQ. 0  ) GO TO 31
      IF(BETAU .GT. 0.) GO TO 102
      WRITE(6,902) BETAU
  102 CONTINUE
      IF(PLOT .EQ. 0)
     +CALL ONESAM(X,N,XO,ON,XS,UXS,CUXS,6,2,NAME,WIDTH,1,U,DHAT,F,FQ,
     +IFIRST,FQWEIB,ORDER,7HWEIBULL,IOPT1,IOPT2,NOBS,FINC,XINT,NQ)
      IFIRST = 0
   31 CONTINUE
      IF(DOH(11).EQ.0)GO TO 32
      IF(PLOT.EQ.0)
     +CALL ONESAM(X,N,XO,ON,XS,UXS,CUXS,6,2,NAME,WIDTH,1,U,DHAT,F,FQ,
     +IFIRST,FQHLOG,ORDER,8HHALF LOG,IOPT1,IOPT2,NOBS,FINC,XINT,NQ)
      IFIRST=0
   32 CONTINUE
  999 CONTINUE
      STOP
  901 FORMAT(//,5X,'SAMPLE SIZE OF ',I5,' G.T. MAX OF 511. RUN ABORTED')
  902 FORMAT(//,5X,F10.4,' IS ILLEGAL VALUE FOR BETA. DATA SET SKIPPED')
  903 FORMAT(20F4.0)
      END
```



```
      FUNCTION AREST(X,L,OPTKHH,OPTCOE)
C
C************************************************************************
C
C     FUNCTION TO COMPUTE AUTOREGRESSIVE ESTIMATOR EVALUATED AT X
C
C     METHOD: ARTUO = OPTKHH / ABS(1 + Y)**2
C             WHERE Y = OPTCOE(J)*EXP(I*J*2*PI*X) SUMMED OVER J = 1, L
C
C     INPUT:
C          X: SCALAR AT WHICH AUTOREGRESSIVE ESTIMATE IS EVALUATED.
C          L: ORDER. MUST BE LESS THAN 11. SEE METHOD.
C          OPTKHH: SEE METHOD.
C          OPTCOE: AUTOREGRESSIVE COEFFICIENTS OF ORDER L. SEE METHOD.
C                  OPTCOE IS A COMPLEX 10-VECTOR.
C
C     OUTPUT: FUNCTION RETURNS VALUE OF AUTOREGRESSIVE ESTIMATOR EVALUATED
C             AT X.
C
C     SUBROUTINES CALLED: NONE.
C
C************************************************************************
C
      COMPLEX OPTCOE(1)
      COMPLEX G
      PI=4.*ATAN(1.0)
      G=CMPLX(1.,0.)
      DO 1 J=1,L
      FJ=J
      G=G+OPTCOE(J)*CEXP(CMPLX(0.,X*2.*PI*FJ))
    1 CONTINUE
      AREST=OPTKHH/REAL(G*CONJG(G))
      RETURN
      END
```

```
      SUBROUTINE AUTORG(A,LS,M,ALPHA,PHI,FKH)
C*****************************************************************
C
C     COMPUTES THE COEFFICIENTS ALPHA(.) AND FKH OF THE
C     AUTOREGRESSIVE ESTIMATOR ACCORDING TO A RECURSIVE
C     ALGORITHM
C
C     INPUT :
C           A : VECTOR OF COMPLEX FOURIER TRANSFORM,
C               OF DIMENSION AT LEAST M
C           M : (M-1) IS THE MAXIMUM ORDER OF SCHEME
C               TO BE COMPUTED
C          LS : ORDER OF SCHEME BEING COMPUTED. LS.GE.1
C
C     OUTPUT :
C          ALPHA : VECTOR OF COEFFICIENTS DEFINING THE
C              APPROXIMATING FUNCTION, HAS TO BE DIMEN-
C              SIONED AT LEAST M AND DECLARED COMPLEX
C          FKH : SCALES THE AUTOREGRESSIVE ESTIMATOR TO
C              INTEGRATE TO A(1), DECLARED REAL
C
C     ALPHA, PHI AND FKH ARE USED RECURSIVELY, IE. THEIR
C     VALUES AT OUTPUT FOR ORDER J ARE USED AS INPUT
C     FOR ORDER (J+1)
C
C*****************************************************************
C
      COMPLEX A(1),ALPHA(1),PHI(1),G,FJH
      COMPLEX FKH
C
      TWOPI=8.*ATAN(1.0)
      FJH=CMPLX(0.,0.)
      PHI(LS) = CMPLX(1.,0.)
      IF(LS .EQ. 1) FKH = CONJG(A(1))
      DO 4 I = 1,LS
    4 FJH=FJH+CONJG(A(I+1)*PHI(I))
      G=-FJH/FKH
      ALPHA(LS) = G
      IF(LS .EQ. 1) GO TO 5
      K = LS -1
      DO 2 I = 1,K
    2 ALPHA(I) = ALPHA(I) + G*PHI(I)
    5 CONTINUE
      DO 3 I=1,LS
    3 PHI(I) = CONJG(ALPHA(LS+1-I))
      FKH=FKH-FJH*CONJG(FJH)/CONJG(FKH)
      RETURN
      END
```



```
      SUBROUTINE DATSTD(X,X0,N,QN)
C*****************************************************************
C
C     SUBROUTINE TO STANDARDIZE VECTOR X BY (X(I)-X0)/(X(N)-X0)
C     AND RETURN THE STANDARDIZED DATA IN THE VECTOR QN
C
C     INPUT :
C           X,N : VECTOR OF ORDERED DATA OF LENGTH N
C           X0 : MINIMUM VALUE TO BE USED IN STANDARDIZATION
C
C     OUTPUT :
C           QN : VECTOR OF STANDARDIZED DATA OF LENGTH N+1
C
C     SUBROUTINES CALLED : NONE
C
C*****************************************************************
C
      DIMENSION X(N),QN(1)
C
      FACT = 1.0/(X(N) - X0)
      QN(1) = 0.0
      DO 10 I = 1,N
   10 QN(I+1)=(X(I) - X0)*FACT
      RETURN
      END



      SUBROUTINE DESTAT(X,N,NAME,IUNIT,IHEAD)
C*************************************************************************
C
C     SUBROUTINE TO PRINT ORDERED ARRAY BY QUANTILES AND COMPUTE
C     DESCRIPTIVE STATISTICS.
C
C     INPUT:
C           X: ARRAY OF ORDER STATISTICS
C           N: DIMENSION OF ARRAY X
C           NAME: NAME OF DATA SET.MUST BE ARRAY OF DIMENSION 20 IN
C                 CALLING PROGRAM.
C           IUNIT: NUMBER OF UNIT OUTPUT IS DESIRED ON.
C
C     OUTPUT: PRINTED OUTPUT IS ON IUNIT.
C
C     NO SUBROUTINES CALLED.
C
C*************************************************************************
C
      DIMENSION X(N),NAME(20),SUM(4),SUMSQ(4)
      DIMENSION ALF(3)
      DIMENSION BOX(5),QUART(5),ITITL(20),L(4)
      DIMENSION IHEAD(20),NAMEY(2),NAMED(2)
      DATA ITITL/4HBOX ,4HPLOT,18*4H    /
      DATA NAMEY/4HQUAR,4HTILE/
      DATA NAMED/4H    ,4H  QN/
      DATA ALF/.05,.10,.25/
C
```

```
C     COMPUTE L,THE ARRAY OF QUARTILE SIZES
C
      LL = N/4
      IERAN = N - 4*LL + 1
      IF(IERAN.EQ.1)GO TO 10
      IF(IERAN.EQ.2)GO TO 11
      IF(IERAN.EQ.3)GO TO 12
      IF(IERAN.EQ.4)GO TO 13
      GO TO 999
C
   10 CONTINUE
      L1 = LL
      L2 = LL
      L3 = LL
      L4 = LL
      GO TO 20
   11 CONTINUE
      L1 = LL
      L2 = LL
      L3 = LL
      L4 = LL + 1
      GO TO 20
   12 CONTINUE
      L1 = LL + 1
      L2 = LL
      L3 = LL
      L4 = LL + 1
      GO TO 20
   13 CONTINUE
      L1 = LL
      L2 = LL + 1
      L3 = LL + 1
      L4 = LL + 1
   20 CONTINUE
      L(1)=L1
      L(2)=L2
      L(3)=L3
      L(4)=L4
C
C
C     PRINT DATA ARRAY - ONE COLUMN FOR EACH QUANTILE.
      WRITE(IUNIT,1001) NAME
      WRITE(IUNIT,1020) IHEAD
      WRITE(IUNIT,1002)
      WRITE(IUNIT,1003)
      DO 30 I = 1,LL
   30 WRITE(IUNIT,1004) I,X(I),X(L1 + I),X(L1+L2+I),
     *                  X(L1+L2+L3+I)
      WRITE(IUNIT,1005)
      IF(L1 .GT. LL) WRITE(IUNIT,1006) X(L1)
      IF(L2 .GT. LL) WRITE(IUNIT,1007) X(L2 + L1 )
      IF(L3 .GT. LL) WRITE(IUNIT,1008) X(L3 + L2 + L1 )
      IF(L4 .GT. LL) WRITE(IUNIT,1007) X(L4 + L3 + L2 + L1 )
      IF(L4 .GT. LL) WRITE(IUNIT,1015) L4
C
```



```
C     COMPUTE AND PRINT DESCRIPTIVE STATISTICS.
C
      K = 1
      KK = 0
      DO 50 I = 1,4
      KK = KK + L(I)
      SUM(I) = 0.0
      SUMSQ(I) = 0.0
      DO 40 J = K,KK
      SUM(I) = SUM(I) + X(J)
      SUMSQ(I) = SUMSQ(I) + X(J)*X(J)
   40 CONTINUE
      K = K + L(I)
   50 CONTINUE
      WRITE(IUNIT,1010) (SUM   (I),I=1,4)
      WRITE(IUNIT,1011) (SUMSQ (I),I=1,4)
      S = SUM(1) + SUM(2) + SUM(3) + SUM(4)
      XBAR = S/FLOAT(N)
      XBI4 = (SUM(2) + SUM(3))/FLOAT(L2 + L3)
      XBO4 = (SUM(1) + SUM(4))/FLOAT(L1 + L4)
      SSQ = SUMSQ(1) + SUMSQ(2) + SUMSQ(3) + SUMSQ(4)
      VAR = (SSQ -S*S/FLOAT(N))/FLOAT(N-1)
      SD = SQRT(VAR)
      R = X(L1+L2+L3+1) - X(L1)
      XMED = (X(N/2 + MOD(N,2)) + X(N/2+1) )/2.
      WRITE(IUNIT,1014) N,XMED             ,XBAR,XBI4,XBO4,VAR,SD,R
      TRIMN = (X(L1)+ 2.*XMED + X(L1+L2+L3+1))/4.
      GASTY = .3*X(N/3+1) + .4*XMED +.3*X(N-N/3)
      WRITE(IUNIT,1019) TRIMN
      WRITE(IUNIT,1016) GASTY
      DO 60 I = 1,3
      IG = INT(ALF(I)*FLOAT(N))
      H = N -2*IG
      NMGM1 = N - IG - 1
      IGP2 = IG + 2
      TRM = X(IG+1)+X(N-IG)
      WMN = X(IG+1)*FLOAT(IG) + X(N-IG)*FLOAT(IG)
      DO 70 J = IGP2,NMGM1
      TRM = TRM + X(J)
      WMN = WMN + X(J)
   70 CONTINUE
      TRM = TRM/H
      WMN = WMN/FLOAT(N)
      WRITE(IUNIT,1017) ALF(I),WMN
      WRITE(IUNIT,1018) ALF(I),TRM
   60 CONTINUE
C
      BOX(1) = X(1)
      BOX(2) = X(L1)
      BOX(3) = XMED
      BOX(4) = X(L1 + L2 + L3 + 1)
      BOX(5) = X(N)
      DO 90 I = 1,5
   90 QUART(I) = FLOAT(I-1)*25.
      CALL JPPLOT(BOX,QUART,5,80,NAMEB,NAMEY,NAME,ITITL,IUPIT,1,2,1,
     *            BOX,IH   ,1,IHEAD)
```

```
  999 CONTINUE
      RETURN
C
 1001 FORMAT(1H1,'DATA SET: ',20A4)
 1002 FORMAT(1H0,20X,'ORDER STATISTICS IN QUARTERS',/1H+,20X,
     +       '_____________________________')
 1003 FORMAT(' SEQUENCE'/'   WITHIN'/' QUARTILE   ',
     +'FIRST QUARTER  SECOND QUARTER   THIRD QUARTER  FOURTH QUARTER'/
     +'+--------  ------------ ------------  ------------ -----
     +________')
 1004 FORMAT(1X,I8 , 4(1X,F15.4))
 1005 FORMAT(1X)
 1006 FORMAT(1H+, 9X,F15.4)
 1007 FORMAT(1H+,25X,F15.4)
 1008 FORMAT(1H+,41X,F15.4)
 1009 FORMAT(1H+,57X,F15.4)
 1010 FORMAT(1H0,'SUM',5X,4(1X,F15.4))
 1011 FORMAT(1H0,'SUM OF'/' SQUARES',1X,4(1X,F15.4))
 1012 FORMAT(1H0,'SUM OF'/' CUBES',3X,4(1X,F15.4))
 1014 FORMAT(1H0,'SAMPLE SIZE = ',I6/' MEDIAN = ',F10.3/
     +' MEAN = ',F10.3/' INNER FOURTHS MEAN = ',F10.3/
     +' OUTER FOURTHS MEAN = ',F10.3/' VARIANCE = ',F10.3/
     +' STANDARD DEVIATION = ',F10.3/' INTERQUARTILE RANGE = ',F10.3)
 1015 FORMAT(1H+,I8)
 1019 FORMAT(1X ,'TRIMEAN = ',F10.3)
 1016 FORMAT(1X ,'GASTWIRTHS ESTIMATE = ',F10.3)
 1017 FORMAT(1X ,F3.2,'-WINSORIZED MEAN = ',F10.3)
 1018 FORMAT(1X ,F3.2,'-TRIMMED MEAN = ',F10.3)
 1020 FORMAT(1X,20A4)
      END
```



```
      SUBROUTINE FORIER(F,U,N,A,MA)
C
C*****************************************************************
C
C   SUBROUTINE TO COMPUTE THE FOURIER TRANSFORM
C   PHI(V) OF A DENSITY DEFINED ON (0,1) FOR V=0,1,...,M
C
C   INPUT :
C          F,U,N : VECTORS OF LENGTH N CONTAINING F(U),U
C          MA : MAXIMUM VALUE OF V FOR WHICH PHI(V) IS COMPUTED
C
C   OUTPUT : A : COMPLEX-VALUED VECTOR CONTAINING THE PHI'S
C
C   SUBROUTINES CALLED : NONE
C
C*****************************************************************
C
      DIMENSION F(1),U(1)
      COMPLEX A(MA),Z
      TWOPI=8.*ATAN(1.)
      FN=FLOAT(N)
      DO 20 IM=1,MA
      FIM=IM-1
      A(IM)=CMPLX(0.,0.)
      DO 10 I=1,N
        Z=CMPLX(0.,TWOPI*FIM*U(I))
   10   A(IM)=A(IM)+F(I)*CEXP(Z)
      A(IM)=A(IM)/FLOAT(N)
   20 CONTINUE
      A(1)=CMPLX(1.,0.)
      RETURN
      END
```

```
      SUBROUTINE KSD(D,U,N,DN,UN,DP,UP)
C
C**************************************************************
C
C     SUBROUTINE TO COMPUTE KOMOLGOROV-SMIRNOFF STATISTIC FOR
C     THE DEVIATIONS D(U)-U.  UPPER AND LOWER BOUNDS ARE GIVEN.
C
C     INPUT : D,U,N
C
C     OUTPUT :
C           DP,UP : MAX (+) DEVIATION, DP, WHICH IS AT U=UP
C           DN,UN : MAX (-) DEVIATION, DN, WHICH IS AT U=UN
C
C     SUBROUTINES CALLED : NONE
C
C**************************************************************
C
      DIMENSION D(N),U(N)
C
      SQN = SQRT(FLOAT(N))
      DP = SQN*(D(1) - U(1))
      UP = U(1)
      DN = DP
      UN = UP
      DO 10   I = 2,N
      DT = SQN*(D(I) - U(I) )
      IF(DT .GT. DP) UP = U(I)
      IF(DT .GT. DP) DP = DT
      IF(DT .LT. DN) UN = U(I)
      IF(DT .LT. DN) DN = DT
   10 CONTINUE
      RETURN
      END
      SUBROUTINE MUHAT(P,X,N,WMEAN,VMEAN,XMEAN)
C****************************************************************
C
C    SUBROUTINE COMPUTES MUHAT BASED ON NORMAL F0OO
C
C    INPUT:  P : TRUNCATION PT
C          X,N : DATA VECTOR OF LENGTH N
C          WMEAN,VMEAN : WEIGHT AND VARIANCE AS DETERMINED BY STEP1
C
C    OUTPUT: XMEAN
C
C    SUBROUTINES CALLED : NONE
C
C****************************************************************
      DIMENSION X(N)
C
      Q=0.
      KP=INT(N*P)
      J=KP+2
      L=N-KP-1
      DO 10 K=J,L
   10 Q=Q+X(K)/FLOAT(N)
      WT1=FLOAT(KP+1)/FLOAT(N)-P
      WT2=1.-P-FLOAT(N-KP-1)/FLOAT(N)
      Q=Q+WT1*X(KP+1)+WT2*X(N-KP)
      XMEAN=(Q+(X(KP+1)+X(N-KP))*WMEAN)*VMEAN
      RETURN
      END
```



```
      SUBROUTINE ONESAM(X,N,X0,QN,XS,WXS,CWXS,IUNIT,IOPT,NAME,
     +NN,JOPT,U,DHAT,F,FQ,IFIRST,FONAME,M,ICASE,IOPT1,IOPT2,NOBS,FINC,
     +XINT,NQ)
C
C**************************************************************
C
C    DRIVER FOR ONE SAMPLE ANALYSIS
C
C    INPUT :
C             X : VECTOR OF SAMPLE OBSERVATIONS
C             N : DIMENSION OF X, XS, WXS, CWXS, U
C            X0 : NATURAL MINIMUM VALUE FOR DATA INPUTTED FROM
C                 MAINLINE
C            NN : NUMBER OF COLUMNS IN PRINTER PLOTS
C          IOPT : 1 OR 2 (POINT OR BAR PLOTS)
C          NAME : NAME OF DATA SET, MUST BE AN ARRAY OF DIMENSION
C                 20 IN CALLING PROGRAM.
C         IUNIT : NUMBER OF UNIT OUTPUT IS DESIRED ON
C          JOPT : IF 0, SCALINGS ARE PLOTTED; IF 1, LOG OF THE
C                 SCALINGS ARE PLOTTED; IF 2, BOTH ARE PLOTTED
C        IFIRST : IS 1 FOR THE FIRST CALL TO ONESAM TO OBTAIN
C                 SAMPLE DIAGNOSTICS AND RAW SPACINGS
C        FONAME : EXTERNAL FUNCTION CORRESPONDING TO
C                 HYPOTHESIZED FO
C             M : MAXIMUM AR ORDER TO BE COMPUTED
C         ICASE : NAME OF HYPOTHESIZED DENSITY IN A-FORMAT
C         IOPT1 : EQUALS 1 FOR GROUPED DATA
C         IOPT2 : EQUALS 1 IF DATA IS TO BE STANDARDIZED TO (0,1)
C          FINC : IS INCREMENT FOR EQUALLY SPACED D(U)
C          XINT : INTERVAL WIDTH FOR GROUPED DATA
C
C    OUTPUT :
C             X : VECTOR OF ORDER STATISTICS (IF IFIRST=1)
C            QN : VECTOR OF STANDARDIZED DATA (IF IFIRST=1)
C            XS : VECTOR OF SPACINGS (IF IFIRST=1)
C           WXS : VECTOR OF WEIGHTED SPACINGS
C          CWXS : VECTOR OF CUMULATIVE WEIGHTED SPACINGS
C          DHAT : VECTOR OF SMOOTHED D'S FOR CASE ICASE
C            FQ : VECTOR OF FQ'S FOR CASE ICASE
C
C    PRINTED OUTPUT IS ON IUNIT.  IT CONSISTS OF ORDER
C    STATISTICS BY QUARTILE, VARIOUS DESCRIPTIVE STATISTICS, AND
C    PRINTER PLOTS OF THE ORDER STATISTICS AND SPACINGS.
C
C    SUBROUTINES CALLED : AUTORG, DATSTD, DESTAT, FORIER, KSD, PARZ,
C          PRINTA, QHLIN, QTOFQ, QUICK, WSPACE, CLPLT1, FCODEA, ICODEA,
C          JPPLOT
C
C    FUNCTIONS CALLED : FONAME, AREST
C
C    EXTERNAL FONAME
C
C**************************************************************
```

```
C
      DIMENSION X(1),XS(1),ON(1),WXS(1),CWXS(1),
     +          FO(1),U(1),IHAT(1),F(1),
     +          CAT(50),RESVEC(50)
      DIMENSION NAME(20),LAB1(20),LAB2(20),LAB3(20),LAB4(20),
     +          LAB5(20),LAB6(20),LAB7(20),LAB8(20),LAB9(20)
      DIMENSION LAB10(20),LAB11(20),LAB12(20)
      DIMENSION LAB13(20),LAB14(20),LAB15(20),LAB16(20)
      DIMENSION NAMON(2),NAMQN(2),NAMU(2),NAMXS(2),NAMUXS(2),
     +          NAMCUS(2),NAMFQ(2),ICASE(2)
      COMPLEX A(50),ALPHA(50),PHI(50),RESVAR
C
      EXTERNAL FGNAME
C
      DATA LAB1/4HRAW ,4HDATA,4H.  N,4HATUR,4HAL N,4HINIM,4HUM: ,
     +4HNONE,12*4H    /
      DATA LAB2/4HRAW ,4HDATA,4H.   ,3*4H    ,4H  SA,4HMPLE,
     +12*4H    /
      DATA LAB3/4HNORM,4HALIZ,4HED D,4HATA ,16*4H    /
      DATA LAB4/20*4H    /
      DATA LAB5/4HPLOT,4H OF ,4HSTAN,4HDARD,4HIZED,4HDATA,
     +14*4H    /
      DATA LAB6/4HPLOT,4H OF ,4HSPAC,4HINGS,16*4H    /
      DATA LAB7/4HSIGM,4HA0 =,4H AVE,4HRACE,4H WEI,4HGHTE,
     +4HD SP,4HACIN,4HG = ,4*4H    ,4HRANG,4HE = ,5*4H    /
      DATA LAB8/4HPLOT,4H OF ,4HRAW ,4HDENS,4HITY ,2*4H    ,
     +4H CAS,4HE   ,11*4H    /
      DATA LAB9/4HD+ =,2*4H    ,4H AT ,4HU = ,2*4H    ,4H.  D,
     +4H- = ,2*4H    ,4H AT ,4HU = ,7*4H    /
      DATA LAB10/4HRAW ,4HDIST,4HTRIB,4HUTIO,4HN = ,4HRAW ,
     +4HD(U),3*4H    ,4H CAS,4HE   ,8*4H    /
      DATA LAB11/4HORDE,4HR = ,18*4H    /
      DATA LAB12/4HIMAT,4H (O),4H PLO,4HTTED,4H AGA,4HINST,
     +4H RAW,4H D (,4H*)  ,3*4H    ,4H CAS,4HE   ,6*4H    /
      DATA LAB13/4HPLOT,4H OF ,4HDENS,4HITY-,4HQUAN,4HTILE,
     +4H FUN,4HCTIO,4HN   ,2*4H    ,4H CAS,4HE   ,7*4H    /
      DATA LAB14/4HPLOT,4H OF ,4HQ(U),4H USI,4HNG L,4HINEA,
     +4HR IN,4HTERP,4HOLAT,4HION ,10*4H    /
      DATA LAB15/4HPLOT,4H OF ,4HRAW ,4HFQ  ,16*4H    /
      DATA LAB16/4HPLOT,4H OF ,4HQ(U),4H USI,4HNG L,4HINEA,
     +4HR IN,4HTERP,4HOLAT,4HION ,4HFOR ,4HGROU,4HPED ,4HDATA,
     +5*4H    /
C
      DATA XONE/.000000001/
      DATA NTAP9/9/
      DATA NAMON,NAMU/4H   Q,4HN(U),4H    ,4H   U/
      DATA NAMXS/4H  GA,4HP(U)/
      DATA NAMUXS,NAMCUS/4HTOGO,4HQ(U),4H    ,4HD(U)/
      DATA NAMIN,NAMFQ/4H IMA,4HT(U),4H   F,4HQ(U)/
C
      NSIGN=N
      N=IABS(N)
      NP1=N+1
      NP2=N+2
      IF(IFIRST.NE.1)GO TO 20
      NG=N
      IF(IOPT1.GE.1)GO TO 120
C
```



```
C     OBTAIN DESCRIPTIVE STATISTICS FOR UNGROUPED DATA
C
      IF(NOBS.EQ.0)NOBS=N
      U(1)=0.
      DO 10 J=1,N
      U(J+1)=(FLOAT(J)-.5)/FLOAT(N)
   10 CONTINUE
      CALL QUICK(N,X)
      IF(XO.NE.XONE)CALL FCODEA(NTAP9,XO,8H(F8.0)  ,LAB1(5))
      IF(NSIGN.GT.0)GO TO 14
      LAB1(12)=ICASE(1)
      LAB1(13)=ICASE(2)
   14 CONTINUE
      CALL DESTAT(X,N,NAME,IUNIT,LAB1)
      XM=X(1)
      IF(XO.NE.XONE)XM=XO
      RANGE=X(N)-XM
      IF(IOPT2.NE.1)GO TO 116
C
C     IF IOPT2=1, STANDARDIZE DATA TO (0,1), OBTAIN DESCRIPTIVE
C               STATISTICS AND, IF FINC.GT.0., COMPUTE Q VIA
C               LINEAR INTERPOLATION
C
      CALL DATSTD(X,XM,N,ON)
      CALL DESTAT(ON(2),N,NAME,IUNIT,LAB3)
      IF(FINC.EQ.0.)GO TO 16
      CALL QHLIN(ON(2),N,FINC,U,FQ,IOPT1,N0,ON(1),XINT)
      NOP1=N0+1
      DO 114 I=1,NOP1
  114 ON(I)=FQ(I)
      DO 15 I=1,4
   15 LAB14(I+10)=LAB3(I)
      DO 115 I=1,20
  115 LAB5(I)=LAB14(I)
   16 NOP1=N0+1
      CALL JFPLOT(ON,U,NOP1,NN,NAMON,NAMU,NAME,LAB5,IUNIT,1,
     +            IOPT,1,ON,1H ,1,LAB4)
      GO TO 19
  116 IF(FINC.EQ.0.)GO TO 18
C
C     IF FINC.GT.0., COMPUTE Q VIA LINEAR INTERPOLATION
C
      CALL QHLIN(X,N,FINC,U,ON,IPT1,N0,XM,XINT)
      NOP1=N0+1
      CALL JPPLOT(ON,U,NOP1,NN,NAMON,NAMU,NAME,LAB14,IUNIT,1,
     +            IOPT,1,ON,1H ,1,LAB4)
      GO TO 19
C
C     IF FINC.EQ.0., COMPUTE Q
C
   18 DO 119 I=1,N
      I2=N-I+2
      I1=N-I+1
  119 ON(I2)=X(I1)
      ON(1)=XM
      CALL JFPLOT(ON,U,NP1,NN,NAMON,NAMU,NAME,LAB1,IUNIT,1,
     +            IOPT,1,ON,1H ,1,LAB4)
C
   19 CONTINUE
C
```

```
      GO TO 20
C
C   COMPUTE B FOR GROUPED DATA
C
  120 CONTINUE
      RANGE=FLOAT(N)*XINT
      CALL GRLIN(X,N,FINC,U,ON,IOPT1,NO,XO,XINT)
      NOP1=NO+1
      CALL JPPLOT(ON,U,NOP1,MM,NAMON,NAMU,NAME,LAB16,IUNIT,1,
     +            IOPT,1,ON,1H ,1,LAB4)
   20 CONTINUE
      IF(NSIGN.LE.0)GO TO 999
C
C   COMPUTE AND PLOT RAW SPACINGS (=O) AND RAW FO (=1./O)
C
      CALL DIGFO(ON,U,NO,XS,FO)
      CALL JPPLOT(XS,U(2),NO,MM,NAMXS,NAMU,NAME,LAB6,IUNIT,1,
     +            IOPT,1,XS,1H ,1,LAB4)
      CALL JPPLOT(FO,U(2),NO,MM,NAMFO,NAMU,NAME,LAB15,IUNIT,1,
     +            IOPT,1,FO,1H ,1,LAB4)
C
C   COMPUTE AND PLOT WEIGHTED SPACINGS FOR CASE ICASE
C
      CALL WSPACE(WXS,CWXS,NO,FO,FONAME,U,SW)
      CALL FCODEA(NTAP9,SW,8H(F10.4) ,LAB7(10))
      CALL FCODEA(NTAP9,RANGE,8H(F10.4) ,LAB7(16))
      LAB8(6)=ICASE(1)
      LAB8(7)=ICASE(2)
      CALL JPPLOT(WXS,U(2),NO,MM,NAMWXS,NAMU,NAME,LAB8,IUNIT,1,
     +            IOPT,1,WXS,1H ,1,LAB7)
C
C   PLOT CUMULATIVE WEIGHTED SPACINGS WITH D+ AND D-
C
      NOP1=NO+1
      CALL KSD(CWXS,U,NO,DM,WM,DP,UP)
      CALL FCODEA(NTAP9,DP,8H(F7.4)  ,LAB9(2))
      CALL FCODEA(NTAP9,UP,8H(F7.4)  ,LAB9(6))
      CALL FCODEA(NTAP9,DM,8H(F7.4)  ,LAB9(10))
      CALL FCODEA(NTAP9,WM,8H(F7.4)  ,LAB9(14))
      LAB10(9)=ICASE(1)
      LAB10(10)=ICASE(2)
      CALL JPPLOT(CWXS,U,NOP1,MM,NAMCWS,NAMU,NAME,LAB10,IUNIT,
     +            1,IOPT,1,CWXS,1H ,1,LAB9)
C
C   COMPUTE FOURIER TRANSFORM OF WEIGHTED SPACINGS
C
      CALL FORIER(WXS,U(2),NO,A,M)
      CALL PRINTA(A,M,IUNIT)
      MM1=M-1
C
C   OBTAIN AUTOREGRESSIVE ESTIMATOR FOR ORDERS 1 TO M-1
C
      DO 30 K=1,MM1
      IORD=K
      CALL ICODEA(NTAP9,IORD,8H(I2)     ,LAB11(3))
      CALL AUTOREG(A,K,M,ALPHA,PHI,RESVAR)
      RESVEC(K)=RESVAR
      NFAC=1
      IF(NO.LE.50)NFAC=4
      NIT=NFAC*NO
      NITP1=NIT+1
```



```
      DO 22 I=1,NITP1
      XI=FLOAT(I-1)/FLOAT(NIT)
      F(I)=AREST(XI,K,RESVAR,ALPHA)
      IF(MOD(I-1,NFAC).NE.0)GO TO 22
      J=(I-1)/NFAC+1
      FO(J)=FONAME(XI)/(F(I)*SW)
   22 CONTINUE
      DHAT(1)=0.
      DO 24 I=2,NOP1
      DHAT(I)=DHAT(I-1)+F(I)/FLOAT(NO)
      IF(NO.GT.50)GO TO 24
      J=NFAC*(I-2)+1
      FWIH=F(J)+4.*F(J+1)+2.*F(J+2)+4.*F(J+3)+F(J+4)
      DHAT(I)=DHAT(I-1)+FWIH/(12.*FLOAT(NO))
   24 CONTINUE
      LAB12(11)=ICASE(1)
      LAB12(12)=ICASE(2)
      CALL JPPLOT(DHAT,U,NOP1,MM,NAMDH,NAMU,NAME,LAB12,IUNIT,1,
     +            1,1,CWXS,1H*,1,LAB11)
      LAB13(10)=ICASE(1)
      LAB13(11)=ICASE(2)
      CALL JPPLOT(FO,U(2),NO,MM,NAMFO,NAMU,NAME,LAB13,IUNIT,1,
     +            IOPT,1,FO,1H ,1,LAB11)
   30 CONTINUE
C
C   COMPUTE 'BEST' ORDER VIA CAT CRITERION AND PRINT
C
      WRITE(IUNIT,901)ICASE
      CALL CLPLT1(RESVEC,M-1,1,4HRVAR,81,1)
      CALL PARZ(RESVEC,M-1,NOBS,CAT,IOPTOR)
      WRITE(IUNIT,902)ICASE
      CALL CLPLT1(CAT,M-1,1,4H CAT,81,1)
      WRITE(IUNIT,903)IOPTOR
C
C
  999 CONTINUE
C
C
  901 FORMAT(///,5X,'RESIDUAL VARIANCES BY AR ORDER',
     +10X,2A4,' CASE'///)
  902 FORMAT(///,5X,'CAT CRITERION',10X,2A4,' CASE'///)
  903 FORMAT(///,5X,'OPTIMAL ORDER = ',I2)
C
      RETURN
      END
```

```
      SUBROUTINE PARZ(RVAR,M,N,CAT,NORD)
C*******************************************************
C
C     SUBROUTINE TO DETERMINE THE ORDER OF AN AUTOREGRESSIVE
C     PROCESS BY PARZENS CAT CRITERIA
C
C     INPUT :
C             N,RVAR(1),...,RVAR(M) : STANDARDIZED RES VAR
C             FOR ORDERS 1 THRU M.
C             N : SAMPLE SIZE
C
C     OUTPUT :
C             NORD : DETERMINED ORDER
C             CAT(1),...,CAT(M)
C
C     SUBROUTINES CALLED : MIN
C
C*******************************************************
C
      DIMENSION RVAR(M),CAT(M)
C
      DN=FLOAT(N)
      DO 1 I=1,M
        C=0.
        DO 2 J=1,I
    2   C=C+((1.-(FLOAT(J)/DN))/RVAR(J))
        C=C/DN
    1 CAT(I)=C-((1.-(FLOAT(I)/DN))/RVAR(I))
      CALL MIN(CAT,M,CATMI,NORD)
      IF(CATMI.GT.-1.) NORD=0
C
      RETURN
      END


      SUBROUTINE PRINTA(A,N,IUNIT)
C
C*******************************************************
C
C     SUBROUTINE TO COMPUTE AND PRINT THE SQUARE MODULUS OF THE
C     COMPLEX-VALUED FOURIER TRANSFORMS A(1),...,A(N)
C
C     INPUT :
C          A,N
C          IUNIT : UNIT ON WHICH OUTPUT IS DESIRED :
C
C     SUBROUTINES CALLED : NONE
C
C*******************************************************
C
      DIMENSION AM(50)
      COMPLEX A(N)
C
      WRITE(IUNIT,1000)
      DO 10 I = 1,N
   10 AM(I) = REAL(CONJG(A(I))*A(I))
      CALL COLPLT(AM,N,1,4HPHI2)
      RETURN
 1000 FORMAT(///,5X,'MODULUS SQUARED OF FOURIER COEFFICIENTS'///)
      END
```



```
      SUBROUTINE QBOX(X,Y,NN1,NAME)
C*******************************************************
C
C     SUBROUTINE OBTAINS PRINTER PLOT OF QUANTILE BOX PLOT WITH
C     MEDIAN LABELED, AND LINES AT HINGES, EIGHTHS, AND 16THS
C
C     INPUT:  X,NN1 : DATA VECTOR OF LENGTH NN1=2*N+1 (IE EXPANDED
C             DATA SET)
C             Y : WORK VECTOR OF LENGTH NN1
C             NAME : VECTOR CONTAINING NAME OF DATA SET
C
C     SUBROUTINES CALLED : NONE
C
C*******************************************************
      DIMENSION X(NN1),Y(NN1),INDU(80),KP(7),NAME(20)
      DATA IB,IM,IX,ID,I2,I3,I4,I5/1H ,1HM,1HX,1H.,1H2,
     +1H3,1H4,1H5/
      NN=NN1-1
      XMIN=X(1)
      RX=X(NN1)-X(1)
C
      DO 10 J=1,NN1
      Y(J)=X(J)
      X(J)=(X(J)-XMIN)/RX
      M=X(J)*79.
      X(J)=80-M
   10 CONTINUE
C
      KP(1)=.0625*FLOAT(NN)+1.
      KP(2)=.125*FLOAT(NN)+1.
      KP(3)=.25*FLOAT(NN)+1.
      KP(4)=.5*FLOAT(NN)+1.
      KP(5)=NN-KP(3)+2
      KP(6)=NN-KP(2)+2
      KP(7)=NN-KP(1)+2
C
C
C
      WRITE(6,1000)NAME
      J=1
      I=1
   12 CONTINUE
      DO 13 K=1,80
   13 INDU(K)=IB
   15 CONTINUE
      JJ=NN1-J+1
      IF(X(JJ).NE.FLOAT(I))GO TO 50
      K=(FLOAT(J-1)/FLOAT(NN))*79
      K=80-K
      DO 20 L=1,7
   20 IF(JJ.EQ.KP(L))GO TO 100
   40 CONTINUE
      IF(INDU(K).EQ.IM)GO TO 45
      IF(INDU(K).EQ.IX)GO TO 140
      IF(INDU(K).EQ.I2)GO TO 142
      IF(INDU(K).EQ.I3)GO TO 144
      IF(INDU(K).EQ.I4)GO TO 146
      IF(INDU(K).EQ.I5)GO TO 146
```

```
      INDU(K)=IX
      GO TO 45
  140 INDU(K)=I2
      GO TO 45
  142 INDU(K)=I3
      GO TO 45
  144 INDU(K)=I4
      GO TO 45
  146 INDU(K)=I5
   45 CONTINUE
      J=J+1
      IF(J.GT.NN1)GO TO 50
      GO TO 15
   50 CONTINUE
      U=FLOAT(JJ)/FLOAT(NN)
      WRITE(6,1005)U,Y(JJ+1),INDU
      I=I+1
      IF(I.GT.80)GO TO 60
      IF(J.GT.NN1)GO TO 40
      GO TO 12
  100 CONTINUE
      IF(L.EQ.4)GO TO 110
      K1=(FLOAT(NN-(J-1))/FLOAT(NN))*79
      K1=80-K1
      KMIN=MIN0(K1,K)
      KMIN=KMIN+1
      KMAX=MAX0(K1,K)
      KMAX=KMAX-1
      DO 102 KK=KMIN,KMAX
  102 INDU(KK)=ID
      IF(INDU(KMIN-1).EQ.IB)INDU(KMIN-1)=ID
      IF(INDU(KMAX+1).EQ.IB)INDU(KMAX+1)=ID
      GO TO 40
  110 INDU(K)=IM
      GO TO 45
   60 CONTINUE
      WRITE(6,1008)
      DO 80 J=1,NN1
   80 X(J)=Y(J)
C
 1000 FORMAT(1H1,/,1X,'DATA SET: ',20A4,/,1X,'QUANTILE-BOX PLOT',/,
     +4X,'U',5X,'Q(U)',/,1X,14(1H-),1X,60(1H-))
 1005 FORMAT(1X,F5.3,F10.3,1X,80A1)
 1008 FORMAT(1X,15(1H-),1X,60(1H-))
C
      RETURN
      END
```



```
      SUBROUTINE QBXPLT(X,NN1,IHEAD,IOPT,U,NBATCH,LAB)
C*****************************************************************
C
C     SUBROUTINE OBTAINS CALCOMP PLOT OF QUANTILE BOX PLOT, WITH
C     IOPT BOXES DRAWN AND A CONFIDENCE INTERVAL FOR THE MEDIAN DRAWN
C
C     INPUT:  X,NN1 : DATA VECTOR OF LENGTH NN1=2*N+1 (IE EXPANDED
C             DATA SET)
C             IHEAD : VECTOR CONTAINS NAME OF DATA SET, USED AS LABEL
C             IOPT : = NUMBER OF BOXES DESIRED DRAWN AT H,E,D,ETC.
C             NBATCH : = NUMBER OF BATCHES ANALYZED.  NBATCH=1 GIVES
C             A SQUARE BOX PLOT 10X10.  NBATCH GT 1 GIVES BOX PLOT 4X10
C             U : WORK VECTOR OF LENGTH NN1
C             LAB : CONTAINS SUBHEADING OF LE 4 CHARACTERS
C
C     SUBROUTINES CALLED : MXMNSC
C     CALCOMP SUBROUTINES CALLED : PLOT, AXIS1, LINE, SYMBOL
C
C*****************************************************************
      DIMENSION X(1),U(1),IHEAD(20),Z(7),UZ(7)
      NN=NN1-1
      NNN1=NN-1
      XMIN=X(1)
      XMAX=X(NN1)
      DO 5 I=1,NNN1
      U(I)=FLOAT(I)/FLOAT(NN)
      X(I)=X(I+1)
    5 CONTINUE
      X(1)=X(2)
      U(1)=U(2)
      CALL PLOT(1.,1.,-3)
      XLEN=10.
      XINC=.1
      IF(NBATCH.GT.1)XLEN=4.0
      IF(NBATCH.GT.1)XINC=.25
      CALL MXMNSC(XMIN,XMAX,10.,A,B)
      CALL AXIS1(0.,0.,'U',-1,XLEN,0.,0.,XINC)
      CALL AXIS1(0.,0.,'Q(U)',4,10.,90.,A,B)
      X(NN)=A
      X(NN1)=B
      U(NN)=0.
      U(NN1)=XINC
      CALL LINE(U,X,NNN1,1,2,4)
      UZ(1)=0.
      UZ(2)=1.
      UZ(3)=0.
      UZ(4)=XINC
      Z(1)=XMIN
      Z(2)=XMAX
      Z(3)=A
      Z(4)=B
      CALL LINE(UZ,Z,2,1,-1,4)
      CALL SYMBOL(.2,10.2,.175,LAB,0.,10)
      CALL SYMBOL(.2,10.5,.28,IHEAD,0.,80)
      IF(IOPT.EQ.0)GO TO 60
      DO 50 J=1,IOPT
      K1=FLOAT(NN)/(2.**(J+1))
      K2=NN-K1
```

```
      UZ(1)=FLOAT(K1)/FLOAT(NN)
      UZ(2)=FLOAT(K2)/FLOAT(NN)
      UZ(3)=UZ(2)
      UZ(4)=UZ(1)
      UZ(5)=UZ(1)
      UZ(6)=0.
      UZ(7)=XINC
      Z(1)=X(K1)
      Z(2)=Z(1)
      Z(3)=X(K2)
      Z(4)=Z(3)
      Z(5)=Z(1)
      Z(6)=A
      Z(7)=B
      CALL LINE(UZ,Z,5,1,0,0)
   50 CONTINUE
   60 CONTINUE
      KH=FLOAT(NN)*.25
      U1=(FLOAT(KH)/FLOAT(NN))*XLEN
      U2=(FLOAT(NN-KH)/FLOAT(NN))*XLEN
      KM=FLOAT(NN)*.5
      XM=(X(KM)-A)/B
      UM=(FLOAT(KM)/FLOAT(NN))*XLEN
      CALL PLOT(U1,XM,3)
      CALL PLOT(U2,XM,2)
      XH=(X(NN-KH)-X(KH))/SQRT(FLOAT(NN))/B
      XMM=XM-XH
      IF(XMM.LT.0.)XMM=0.
      XMP=XM+XH
      IF(XMP.GT.10.)XMP=10.
      CALL PLOT(UM,XMM,3)
      CALL PLOT(UM,XMP,2)
      CALL PLOT(-1.,-1.,-3)
      DO 70 I=1,NNM1
      IM=NN-I
   70 X(IM+1)=X(IM)
      X(1)=XMIN
      X(NN1)=XMAX
      XLEN1=XLEN+1.
      CALL PLOT(XLEN1,0.,-3)
      RETURN
      END
```



```
      SUBROUTINE ODIAG(X,N,IOPT,NAME)
C**************************************************************
C
C     SUBROUTINE ORDERS, PRINTS DATA, EXPANDS DATA SET BY COMPUTING
C     AVERAGES OF CONSECUTIVE DATA PTS AND COMPUTES DIAGNOSTICS
C
C     INPUT:  X,N : DATA VECTOR OF LENGTH N.  MUST BE DIMENSIONED
C             AT LEAST 2*N+1.  ON OUTPUT, X CONTAINS THE EXPANDED
C             DATA SET
C             IOPT : =1 IF USER WISHES TO INPUT A NATURAL MIN IN
C             X(N+1) AND NATURAL MAX IN X(N+2), EG WHEN USING
C             SEVERAL BATCHES
C             NAME : VECTOR CONTAINING NAME OF DATA SET USED FOR LABEL
C
C     SUBROUTINES CALLED : QUICK, XBARS2(TSBD), MORRIS(IMSL)
C
C**************************************************************
      DIMENSION X(1),NAME(20)
      WRITE(6,1001)NAME,N
      WRITE(6,1002)(X(I),I=1,N)
      CALL XBARS2(X,N,XBAR,S2)
      SD=SQRT(S2)
      IF(IOPT.EQ.1)XMIN=X(N+1)
      IF(IOPT.EQ.1)XMAX=X(N+2)
      CALL QUICK(N,X)
      WRITE(6,1009)
      WRITE(6,1002)(X(I),I=1,N)
      NN=2*N
      NN1=NN+1
      NM1=N-1
      DO 10 J=1,N
      JM1=J-1
      JNE=2*(N-JM1)
      JNO=2*N-2*JM1-1
      NJ=N-JM1
      NJ1=N-JM1-1
      IF(NJ1.EQ.0)NJ1=1
      X(JNE)=X(NJ)
      X(JNO)=(X(NJ)+X(NJ1))/2.
   10 CONTINUE
      X(NN1)=X(NN)
      IF(IOPT.EQ.1)X(1)=XMIN
      IF(IOPT.EQ.1)X(NN1)=XMAX
      K1=.0625*FLOAT(NN)
      K2=.125*FLOAT(NN)
      K3=.25*FLOAT(NN)
      K4=.5*FLOAT(NN)
      K5=NN-K3
      K6=NN-K2
      K7=NN-K1
      WRITE(6,1008)
      WRITE(6,1003)K7,X(K7+1),K6,X(K6+1),K5,X(K5+1),K4,X(K4+1),K3,
     *X(K3+1),K2,X(K2+1),K1,X(K1+1)
      HMID=(X(K5+1)+X(K3+1))/2.
      EMID=(X(K6+1)+X(K2+1))/2.
      DMID=(X(K7+1)+X(K1+1))/2.
      AVMID=(X(K4+1)+HMID+EMID+DMID)/4.
      HHR=(X(K5+1)-X(K3+1))/SQRT(FLOAT(N))
```

```
      WRITE(6,1004)X(K4+1),HMID,EMID,DMID,AVMID,XBAR,HHN
      CALL MDNRIS(.75,DENH,IER)
      XHNOR=(X(K5+1)-X(K3+1))/(2.*DENH)
      CALL MDNRIS(.875,DENE,IER)
      XENOR=(X(K6+1)-X(K2+1))/(2.*DENE)
      CALL MDNRIS(.9375,DEND,IER)
      XDNOR=(X(K7+1)-X(K1+1))/(2.*DEND)
      AVNOR=(XHNOR+XENOR+XDNOR)/3.
      XHEXP=(X(K5+1)-X(K3+1))/ALOG(3.)
      XEEXP=(X(K6+1)-X(K2+1))/ALOG(7.)
      XDEXP=(X(K7+1)-X(K1+1))/ALOG(15.)
      AVEXP=(XHEXP+XEEXP+XDEXP)/3.
      WRITE(6,1005)XHNOR,XHEXP,XENOR,XEEXP,XDNOR,XDEXP,AVNOR,AVEXP,SD
      XMH=(X(K4+1)-X(K3+1))/(X(K5+1)-X(K3+1))
      XME=(X(K4+1)-X(K2+1))/(X(K6+1)-X(K2+1))
      XMD=(X(K4+1)-X(K1+1))/(X(K7+1)-X(K1+1))
      WRITE(6,1006)XMH,XME,XMD
      XHE=ALOG((X(K5+1)-X(K3+1))/(X(K6+1)-X(K2+1)))
      XHD=ALOG((X(K6+1)-X(K2+1))/(X(K7+1)-X(K1+1)))
      XHEN=ALOG(DENH/DENE)
      XHDN=ALOG(DENE/DEND)
      XHEE=ALOG(ALOG(3.)/ALOG(7.))
      XHDE=ALOG(ALOG(7.)/ALOG(15.))
      WRITE(6,1007)XHE,XHEN,XHEE,XHD,XHDN,XHDE
      RETURN
 1001 FORMAT(1H1,/,1X,'DATA SET: ',20A4,/,1X,'SAMPLE SIZE = ',I5,//)
 1002 FORMAT(5X,5F12.3)
 1003 FORMAT(//,1X,50(1H*),//,1X,'7 POINT SUMMARY',/,
     +5X,'UPPER D      = X(',I4,') =',F12.3,/,
     +5X,'UPPER E      = X(',I4,') =',F12.3,/,
     +5X,'UPPER HINGE  = X(',I4,') =',F12.3,/,
     +5X,'MEDIAN       = X(',I4,') =',F12.3,/,
     +5X,'LOWER HINGE  = X(',I4,') =',F12.3,/,
     +5X,'LOWER E      = X(',I4,') =',F12.3,/,
     +5X,'LOWER D      = X(',I4,') =',F12.3,//)
 1004 FORMAT(1X,'MIDSUMMARIES',/,2X,'MEDIAN = ',F12.3,/,
     +2X,'MIDH   = ',F12.3,/,2X,'MIDE   = ',F12.3,/,
     +2X,'MIDD   = ',F12.3,/,2X,'AVE    = ',F12.3,//,
     +2X,'XBAR   = ',F12.3,//,2X,'HH/SQRT(N) = ',F12.3,//)
 1005 FORMAT(1X,'SCALE SUMMARIES',/,
     +2X,'HH/HH(NORMAL) = ',F12.3,2X,'HH/HH(EXP) = ',F12.3,/,
     +2X,'EE/EE(NORMAL) = ',F12.3,2X,'EE/EE(EXP) = ',F12.3,/,
     +2X,'DD/DD(NORMAL) = ',F12.3,2X,'DD/DD(EXP) = ',F12.3,/,
     +2X,'AVE           = ',F12.3,2X,'AVE        = ',F12.3,//,
     +2X,'SAMPLE ST DEV = ',F12.3,//)
 1006 FORMAT(1X,'SKEWNESS DIAGNOSTIC MEASURES',/,2X,'MH/HH = ',F7.4,/,
     +2X,'ME/EE = ',F7.4,/,2X,'MD/DD = ',F7.4,//)
 1007 FORMAT(1X,'TAIL DIAGNOSTIC MEASURES',/,20X,'SAMPLE',6X,'NORMAL(0,1
     +)',4X,'EXP(1)',/,2X,'LOG(HH/EE) = ',3F12.4,/,2X,'LOG(EE/DD) = ',
     +3F12.4,//,1X,50(1H*),//)
 1008 FORMAT(1H1)
 1009 FORMAT(//,1X,'ORDERED DATA',//)
      END
```



```
      SUBROUTINE QILIN(X,N,H,U,Q,IOPT,NQ,XMIN,XINT)
C*********************************************************************
C
C     SUBROUTINE TO COMPUTE Q(U) AT U = H,2H,3H,...,1-H WHERE
C     H IS > OR = .002 BY LINEAR INTERPOLATION FROM ORDER
C     STATISTICS IF DATA IS UNGROUPED (IOPT=0), OR FROM TALLIES
C     IF DATA IS GROUPED (IOPT=1).
C
C     INPUT :
C
C             X : VECTOR OF ORDER STATISTICS IF DATA IS UNGROUPED
C                 VECTOR OF TALLIES IF DATA IS GROUPED
C             N : DIMENSION OF X
C          XMIN : NATURAL MINIMUM (IF DATA IS UNGROUPED)
C                 LOWER LIMIT OF 1ST GROUP (FOR GROUPED DATA)
C          XINT : GROUP WIDTH (IF DATA IS GROUPED)
C          IOPT : =0 IF DATA IS UNGROUPED
C                 =1 IF DATA IS GROUPED
C             H : FIRST VALUE (AND INCREMENT) FOR COMPUTING Q
C                 USUALLY H=.01,.025, OR .05
C
C     OUTPUT :
C
C             U : VECTOR OF U VALUES (MAX 500)
C             Q : VECTOR CONTAINING Q(U) (MAX 500)
C
C     SUBROUTINES CALLED : NONE
C
C*********************************************************************
C
      DIMENSION X(1),U(1),Q(1)
C
      NQ=1./H
      IF(NQ.GT.500)GO TO 900
      IF((FLOAT(NQ)*H).EQ.1.) NQ=NQ-1
      IF(IOPT.EQ.1)GO TO 50
C
C     COMPUTE Q FOR UNGROUPED DATA
C
      X(N+1)=X(N)
      DO 10 I=1,NQ
      J=FLOAT(N)*FLOAT(I)*H+.5
      U(I)=FLOAT(I)*H
      A=U(I)*FLOAT(N)-FLOAT(J)+.5
      Q(I)=A*X(J+1)+(1.-A)*X(J)
   10 CONTINUE
      GO TO 80
   50 CONTINUE
C
C     COMPUTE Q FOR GROUPED DATA
C
      XDEN=0.
      DO 60 I=1,N
      XDEN=XDEN+X(I)
   60 CONTINUE
      UL=0.
      XNUM=0.
      IU=1
      IV=1
      U(1)=H
   65 XNUM=XNUM+X(IU)
      UU=XNUM/XDEN
```

```
   70 IF(UL.LE.U(IV).AND.U(IV).LT.UU)GO TO 75
      IU=IU+1
      UL=UU
      GO TO 65
   75 CONTINUE
      A=(U(IV)-UL)/(UU-UL)
      XL=XMIP+XINT*FLOAT(IU-1)
      XU=XL+XINT
      Q(IV)=A*XU+(1.-A)*XL
      IV=IV+1
      IF(IV.GT.M)GO TO 80
      U(IV)=FLOAT(IV)*H
      GO TO 70
   80 CONTINUE
      DO 90 I=1,NQ
      I2=NQ-I+2
      I1=I3-I+1
      Q(I2)=Q(I1)
   90 U(I2)=U(I1)
      Q(1)=XMIN
      U(1)=0.
      GO TO 999
C
  900 WRITE(6,901)H
  901 FORMAT(////,5X,'INCREMENT OF',F5.4,' IS TOO SMALL TO COMPUTE',
     +' Q',/'INCREMENT MUST BE AT LEAST .002')
C
  999 CONTINUE
C
      RETURN
      END

      SUBROUTINE QTOFQ(Q,U,NQ,XS,FQ)
C*************************************************************
C
C     SUBROUTINE TO COMPUTE Q(U) AND FQ(U)=1./Q(U) FROM
C     THE EMPIRICAL QUANTILE FN CAP Q(U) AND THE U VALUES.
C
C     INPUT : Q,U,NQ
C     OUTPUT : XS : SPACINGS-Q(U)
C              FQ
C
C*************************************************************
C
      DIMENSION Q(1),U(1),XS(NQ),FQ(NQ)
C
      XS(NQ)=(Q(NQ+1)-Q(NQ))/(U(NQ+1)-U(NQ))
      XSMIN=XS(NQ)
      NQM1=NQ-1
      DO 10 I=1,NQM1
      XS(I)=(Q(I+2)-Q(I))/(U(I+2)-U(I))
      IF(XS(I).EQ.0.0)GO TO 10
      IF(XS(I).LT.XSMIN)XSMIN=XS(I)
   10 CONTINUE
      XSMIN=.25*XSMIN
      DO 20 I=1,NQ
      IF(XS(I).EQ.0.0)XS(I)=XSMIN
      FQ(I)=1./XS(I)
   20 CONTINUE
      RETURN
      END
```



```
      SUBROUTINE QUICK(N,T)
C**********************************************************************
C
C     QUICK SORT.THIS ALGORITH IS ALSO REFERRED TO AS A PARTITIONED
C     EXCHANGE SORT.EXPECTED RUNTIME IS PROPORTIONAL TO N*LOG2(N)
C     ALTHOUGH THE WORST CASE IS PROPORTIONAL TO N**2.
C     REFERENCE: DONALD E. KNUTH- THE ART OF COMPUTER PROGRAMMING VOL 3.
C
C     INPUT :
C           X,N : VECTOR TO BE SORTED OF LENGTH N
C
C     OUTPUT :
C           X : SORTED VECTOR
C
C     SUBROUTINES CALLED : NONE
C
C**********************************************************************
C
      REAL T(N),Y
      INTEGER IP,LV(16),IV(16),LP,IUP
C
      LV(1)=1
      IV(1)=N
      IP=1
   10 IF(IP.LT.1) GO TO 75
   15 IF((IV(IP)-LV(IP)).LT.1) GO TO 20
      GO TO 25
   20 IP=IP-1
      GO TO 10
   25 LP=LV(IP)-1
      IUP=IV(IP)
      Y=T(IUP)
   30 IF((IUP-LP).LT.2) GO TO 45
      LP=LP+1
      IF(T(LP).LE.Y) GO TO 30
      T(IUP)=T(LP)
   35 IF((IUP-LP).LT.2) GO TO 40
      IUP=IUP-1
      IF(T(IUP).GE.Y) GO TO 35
      T(LP)=T(IUP)
      GO TO 30
   40 IUP=IUP-1
   45 T(IUP)=Y
      IF((IUP-LV(IP)).LT.(IV(IP)-IUP)) GO TO 55
      GO TO 60
   55 LV(IP+1)=LV(IP)
      IV(IP+1)=IUP-1
      LV(IP)=IUP+1
      GO TO 70
   60 LV(IP+1)=IUP+1
      IV(IP+1)=IV(IP)
      IV(IP)=IUP-1
   70 IP=IP+1
      GO TO 15
   75 RETURN
      END
```

```
      SUBROUTINE SIGHAT(P,Y,N,USIGMA,VSIGMA,SIGMA)
C*****************************************************************
C
C   SUBROUTINE COMPUTES SIGMAHAT BASED ON NORMAL FQ00
C
C   INPUT:  P : TRUNCATION PT
C           Y,N : DATA VECTOR OF LENGTH N
C           USIGMA,VSIGMA : WEIGHT AND VARIANCE AS COMPUTED BY STEP1
C
C   OUTPUT: SIGMA
C
C   SUBROUTINES CALLED : FQNORM
C
C*****************************************************************
C
      DIMENSION Y(N)
C
      QPHI=0.
      KP=INT(N*P)
      J=KP+2
      L=N-KP-1
      DO 10 K=J,L
      X=FLOAT(K-1)/FLOAT(N)
      Y2=FLOAT(K)/FLOAT(N)
      X1=FQNORM(X)
      Y1=FQNORM(Y2)
   10 QPHI=Y(K)*(X1-Y1)+QPHI
      Z1=FLOAT(KP+1)/FLOAT(N)
      WT1=FQNORM(P)-FQNORM(Z1)
      Z2=FLOAT(N-KP-1)/FLOAT(N)
      WT2=FQNORM(Z2)-FQNORM(1.-P)
      QPHI=QPHI+WT1*Y(KP+1)+WT2*Y(N-KP)
      SIGMA=(2.*QPHI+(Y(KP+1)-Y(N-KP))*USIGMA)*VSIGMA
      RETURN
      END
      SUBROUTINE STEP1(P,WMEAN,USIGMA,VMEAN,VSIGMA)
C*****************************************************************
C
C   SUBROUTINE COMPUTES WEIGHTS AND VARIANCE FOR TRUNCATION PT P
C   TO BE USED IN FINDING MUHAT AND SIGHAT BASED ON NORMAL FQ00
C
C   INPUT : P
C
C   OUTPUT: WMEAN,USIGMA,VMEAN,VSIGMA
C
C   SUBROUTINES CALLED : MDNRIS(IMSL),FQNORM
C
C*****************************************************************
      IF(P.EQ.0.)GO TO 20
      CALL MDNRIS(P,T,IER)
      FQ00=FQNORM(P)
      WMEAN=FQ00*(T+FQ00/P)
      USIGMA=T*WMEAN-FQ00
      VMEAN=1./(1.-2.*P+2.*WMEAN)
      VSIGMA=1./(2.*(1.-2.*P+2.*T*FQ00*T*USIGMA))
      GO TO 30
   20 CONTINUE
      WMEAN=0.
      USIGMA=0.
      VMEAN=1.
      VSIGMA=.5
   30 CONTINUE
      RETURN
      END
```



```
      SUBROUTINE USPACE(UXS,CUXS,NO,FO,FQ00,U,SIGO)
C
C*****************************************************************
C
C   SUBROUTINE TO COMPUTE D(U), CUMULATIVE D'S, AND SIGMAO
C   FOR THE MODEL D(U)=MU+SIGMA*DO(U)
C
C   INPUT :
C          FO,NO : VECTOR OF LENGTH NO CONTAINING F(U)
C          FQ00 : HYPOTHESIZED DENSITY QUANTILE FN--EXTERNAL
C                 FUNCTION
C          U : VECTOR OF LENGTH NO CONTAINING U VALUES
C
C   OUTPUT :
C          UXS : VECTOR OF LENGTH NO CONTAINING D(U)
C          CUXS : VECTOR OF LENGTH NO CONTAINING THE
C                 CUMULATIVE D'S
C          SIGO : COMPUTED VALUE OF SIGMAO = CUXS(NO)
C
C   SUBROUTINES CALLED : NONE
C
C*****************************************************************
C
      DIMENSION FO(1),U(1),UXS(NO),CUXS(1)
C
      NOP1=NO+1
      CUXS(1)=0.
      DO 10 I=2,NOP1
      FAC=FQ00(U(I))
      UXS(I-1)=FAC/FO(I-1)
      CUXS(I)=CUXS(I-1)+UXS(I-1)
   10 CONTINUE
      SIG00=CUXS(NOP1)
      SIGO=SIG00/FLOAT(NO)
      DO 20 I=1,NO
      UXS(I)=UXS(I)/SIGO
      CUXS(I)=CUXS(I)/SIG00
   20 CONTINUE
      CUXS(NOP1)=1.
      RETURN
      END
```